CS281B/Stat241B: Advanced Topics in Learning & Decision Making

Soft Margin SVM

Lecturer: Michael I. Jordan *Scribe: Anat Caspi*

1 SVM Non-separable Classification

We return to our discussion of classification, this time addressing the problems of nonseparable datasets, as well as sensitivity to outliers. The key is in the introduction of slack variables (see optimization techniques for more details on slack variables). Whereas before, our margins were defined by the hard constraint

$$y_i(x_i^T w + b) \geq 1 \quad y_i \epsilon \{-1, 1\} \tag{1}$$

in the problem posed here, we soften the constraint:

$$y_i(x_i^T w + b) \geq 1 - \xi_i \qquad \xi_i \geq 0 \tag{2}$$

The new constraint permits a functional margin that is less than 1, and contains a penalty of cost $C\xi_i$ for any data point that falls within the margin on the correct side of the separating hyperplane (i.e., when $0 < \xi_i \leq 1$), or on the wrong side of the separating hyperplane (i.e., when $\xi_i > 1$). We thus state a preference for margins that classify the training data correctly, but soften the constraints to allow for non-separable data with a penalty proportional to the amount by which the example is misclassified. The parameter C controls the relative weighting between the goal of making the margin small and ensuring that most examples have functional margins that are at least 1. We would therefore like to minimize the sum total of the penalties ξ_i over all i (this is an upper bound on the training classification errors). Thus, the new well-posed optimization problem (using l_1 regularization) becomes:

$$\begin{aligned} min \qquad & \frac{1}{2}w^T w + C\sum_{i=1}^{m} \xi_i \\ s.t. \qquad & y_i(x_i^T w + b) \geq 1 - \xi_i \qquad \xi_i \geq 0 \end{aligned} \tag{3}$$

In practice, C is determined by cross validation. We will not address the actual best value for C. Note only that using 1 by default does not work very well, while $\frac{1}{n}$ may work better.

Ideally, we are interested in the number of non zero ξ_i, as that is the count of errors made by our classifier on the set of training examples. This count is the L_0 norm, and is discontinuous and nonconvex. We would like to stay close to L_0, while maintaining convexity. Refer to [1] on zero-norm optimization.

We take the lagrangian in the usual manner:

$$\mathcal{L}(w, \xi, b, \alpha) = \frac{1}{2}w^T w + C\sum_{i=1}^{m} \xi_i + \sum_{i=1}^{m} \alpha_i \left[1 - \xi_i - y_i(x_i^T w + b)\right] \tag{4}$$

[1] Weston J., Elisseeff A., Scholkopf B., Tipping M., Use of the Zero-Norm with Linear Models and Kernel Methods, JMLR 3 (2003) 1439-1461.

To find the dual form of the problem, we first need to minimize $\mathcal{L}(w, \xi, b, \alpha)$ with respect to w, ξ, and b (for fixed α), to get Θ_D.

$$\begin{array}{c} min \\ w, \xi, b \end{array} \mathcal{L}(w, \xi, b, \alpha), \quad \xi_i \geq 0 \tag{5}$$

Since the Lagrangian function is linear in α, we cannot set the gradient with respect to α to zero. We obtain the following dual optimization problem:

$$D: \begin{array}{c} max \\ \alpha \end{array} \qquad \Theta(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y_i y_j \alpha_i \alpha_j \left\langle \phi(x^{(i)}), \phi(x^{(j)}) \right\rangle. \tag{6}$$

$$s.t. \qquad 0 \leq \alpha_i \leq C \tag{7}$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0 \tag{8}$$

The introduction of the box constraint on α_i is necessary to ensure that the Lagrangian will be bounded (i.e., that we cannot drive the cost to $-\infty$). This will occur in the case in which $(C - \alpha_i)$ is negative and ξ_i goes to ∞, the summed expression $\sum_{i=1}^{m}(C - \alpha_i)\xi_i$ goes to $-\infty$. The box constraint prevents this from happening.

The dual still has a quadratic objective, and differs from the optimal margin classifier only in the introduction of the box constraint. Indeed, we can still use

$$w = \sum_{i=1}^{m} \alpha_i x_i^T y_i \tag{9}$$

to give us the optimal value of w in terms of the optimal value of α.

We must also verify that the the KKT dual-complementarity conditions are still satisfied in this optimization problem:

$$\alpha_i = 0 \qquad \Rightarrow y_i \left[x_i^T w + b \right] \geq 1 \tag{10}$$

$$\alpha_i = C \qquad \Rightarrow y_i \left[x_i^T w + b \right] \leq 1 \tag{11}$$

$$0 < \alpha_i < C \qquad \Rightarrow y_i \left[x_i^T w + b \right] = 1 \tag{12}$$

As before, α_i will be nonzero only for the support vectors, where the set of support vectors now includes all data points on the margin boundary as well as those on the wrong side of the margin boundary.

2 The Sequential Minimal Optimization Algorithm

The Sequential Minimal Optimization (SMO) algorithm [2] introduced by John Platt provides an efficient algorithm for solving the dual problem. The dual optimization problem we wish to solve is stated in (6),(7), (8). This can be a very large QP optimization problem. Standard interior point methods (such as the projected conjugate gradient algorithm) attempt to optimize each move in the entirety of dual space. SMO

[2] John Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schlkopf, C. Burges, and A. Smola, editors, Advances in Kernel Methods - Support Vector Learning. MIT Press, 1998.

breaks the large QP problem into small subsets (a series of the smallest possible QP sub-problems including only two α's at a time), which can be solved analytically. This approach is up to an order of magnitude faster. Without kernel caching, SMO scales somewhere between linear and quadratic in the training set size, while a standard interior point method scales somewhere between linear and cubic in the training set size.

We may want to question the motivation for solving the dual over the primal. We note that the primal has arbitrary linear constraints, while the dual has the anticipated box constraints. The dual is therefore nicer to solve.

3 The General Margin

For a generalized notion of the margin, we can define the margin in terms of a boundary function $f(x_i)$.

$$margin = y_i f(x_i) \tag{13}$$

Where in the linear formulation, $f(x_i)$ is defined as:

$$f(x_i) = w^T \phi(x_i) + b. \tag{14}$$

The size of the margin measures some distance metric from the boundary line, whereas the classification is based on thresholding at the boundary. That is, we classify an example as correct if the sign of y_i is the same as the sign of $f(x_i)$, and incorrect otherwise. Since noise or inseparable data make it impossible (or undesirable) to find a hard margin (that is, a weight vector that satisfies all examples), we introduce a loss function, L, to measure the magnitude of the error yielded by a particular weight vector, w. For example, the $0-1$ loss function, given by $L(Y, f(X)) = \sum_{i=1}^{n} H(-y_i f(x_i))$ where $H(x)$ is 1 if $x \geq 0$ and 0 otherwise. This loss function is a natural objective function as it sums the number of errors made over the training set. However, this is not a convex function, and the problem can be shown to be NP-hard. We could try to relax this to a convex problem by decreasing the upper bound.

Claim: The soft-margin SVM is a convex program for which the objective function is the hinge loss.

Proof: We have the original problem as stated in (3) with the regularizer ($w^T w$) and the loss ($C \sum_{i=1}^{m} \xi_i$). We can move the linear constraints into the objective function as follows:

$$min \qquad w^T w + C \sum_{i=1}^{m} (1 - y_i f(x_i))_+ \tag{15}$$

$$where \qquad z_+ = \max(z, 0) \tag{16}$$

While the term $(1 - y_i(f(x_i))$ grows more negative as a function of error, the loss itself is capped at 0 (when the value of $y_i f(x_i) \geq 1$). So while we're trying to minimize the sum of the losses, we are in fact minimizing the hinge loss. This is an unconstrained non-quadratic problem with a more complex objective function, but it is piece-wise linear and convex. Though there is some advantage to driving the margin larger (i.e., not capping it at 0), the discontinuity of the hinge loss leads to support vectors, so it is overall good but convergence is not guaranteed with steepest descent methods.

The log logistic loss function is a smooth function that is similar to the hinge loss. It too pushes down as an upper bound to the $0 - 1$ loss. It does not have the piece-wise linear property, so to do logistic regression we use a stepwise Newton method which (unlike the case of the hinge loss) guarantees convergence.

Other proposed loss functions are shown in Figure 1. In addition, it was noted that Adaboost uses an exponential loss function.

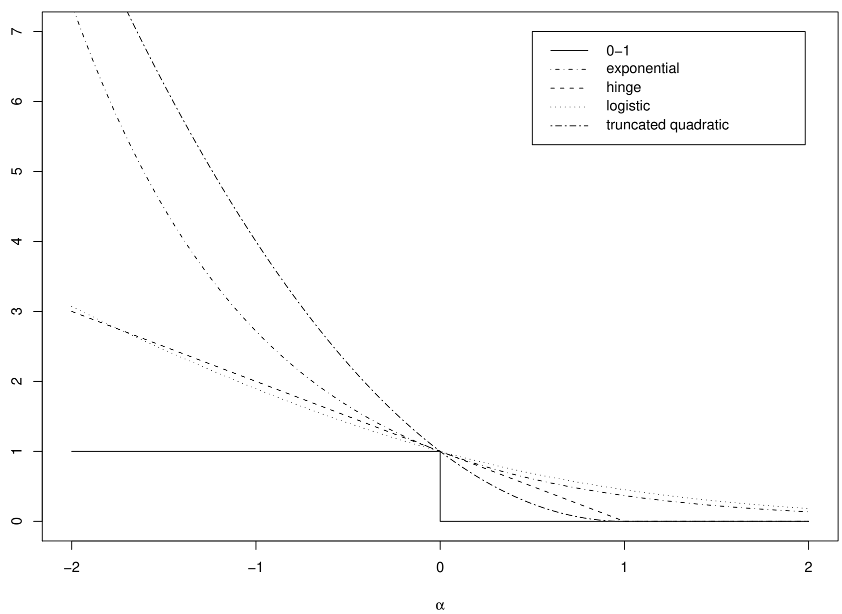

Figure 1: Various loss functions that upper bound the one-zero loss.